الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research
National School of Cybersecurity
Department of Foundation Training

وزارة التعليم العالي والبحث العلمي
المدرسة الوطنية العليا في الأمن السيبراني
قسم التكوين القاعدي

Material: Algorithms and Dynamic Data Structures

Academic Year: 2024/2025 **Level:** 1st Year

Tutorial/Practical work 03: Queues & Stacks (2/2)

Exercise 1: Reverse a Stack Using Another Stack (Dynamic Implementation)

Write an algorithm to reverse the contents of a stack using only Push and Pop operations and an auxiliary stack.

Exercise 2: Evaluate a Postfix Arithmetic Expression Using a Stack (Dynamic Implementation)

Given a postfix expression (e.g., 5 6 2 + * 12 4 / -), write an algorithm that evaluates it using a dynamic stack implemented via a linked list.

Exercise 3: Check Balanced Parentheses Using a Stack

Write an algorithm that uses a stack (dynamic implementation) to verify if a given expression contains balanced parentheses.

Exercise 4: Convert Infix Expression to Postfix (Reverse Polish Notation)

Write a C program that takes an infix arithmetic expression (e.g., A + B * C) and converts it to a postfix expression (A B C * +) using a stack (implemented dynamically with linked lists).

Exercise 5: Implement Undo/Redo System Using Two Stacks

Simulate a basic Undo/Redo mechanism using two stacks:

- UndoStack
- RedoStack

A user performs a sequence of actions like writing letters ('A', 'B', 'C', etc.).
You need to:

- Store each new action in the Undo stack

On Undo:

- Pop from Undo and push into Redo

On Redo:

- Pop from Redo and push back into Undo